Principal Components of Orthogonal Object-Oriented Metrics

Victor Laing
SRS Information Services
Software Assurance Technology Center
http://satc.gsfc.nasa.gov/

Problem

- **How can we do More with Less?**
  - Hundreds of software metrics for traditional and object-oriented software
  - Information overload too time consuming for mangers and developers to interpret

- **Goals of the Research**
  - Find a easy-to-use minimum set of software metrics capable of measuring the overall quality of object oriented software
  - Assumption: higher object-orientedness ⇒ higher quality software

Software Metrics

- **Some Traditional Metrics**
  - McCabe Cyclomatic Complexity (CC)
  - Source Lines of Code (SLOC)
  - Comment Percentage (CP)

- **Some Object-Oriented Metrics**
  - Weighted Methods Per Class (WMC)
  - Coupling Between Object Classes (CBO)
  - Response for a Class (RFC)
  - Depth of Inheritance Tree (DIT)
  - Lack of Cohesion in Methods (LOCM)

Software Metrics

- **Results of Previous Research**
  - Traditional metrics do not capture certain fundamental aspects of the object-oriented paradigm
  - Chidamber and Kemerer: CK Metrics Suite, 1994
  - CK suite validated by Basili in 1996 and again by Tang in 1999
  - Many other object-oriented metrics are derived from the CK suite of object-oriented metrics

Current Research

- **Object-Oriented Metrics for this Study**
  - Weight Methods per Class (WMC)
  - Coupling Between Object Classes (CBO)
  - Response for a Class (RFC)

- **Definitions**
  - RFC = NLM + NRM
  - NLM = number of local methods in a class
  - NRM = number of remote methods called
  - Let NLM = WMC

Current Research

- **Under Tight Coupling of Objects**
  - NRM $\approx$ CBO
  - Substituting WMC and CBO for NLM and NRM respectively in RFC = NLM + NRM gives RFC = WMC + CBO
- **Results of the Current Research**
  - Approximately RFC = WMC + CBO
  - This simple equation measures the object-orientedness of object-oriented software

Empirical Investigation

- **Projects**
  - Project A: 46 Java Classes (Commercial Software)
  - Object-Oriented Constructs

  - Project B: 1000 Java Classes (NASA Software)
  - Excellent Object-Oriented Constructs

  - Project C: 1617 C++ Classes (NASA Software)
  - Good Object-Oriented Constructs

Data Analysis

Table 3: Correlation of Metrics over Projects

	CBOxRFC	CBOxWMC	RFCxWMC
Project A	0.83	0.43	0.47
Project B	0.28	0.11	0.75
Project C	0.35	0.26	0.83

Metric 1 x Metric 2 = correlation between metric 1 and metric 2.

Data Analysis

- **Regression Model**

  - $RFC = \beta_1 WMC + \beta_2 CBO + Constant$

Table 4: Regression Models

	β_{WMC}	β_{CBO}	**Constants**	R^2
RFC_A	0.131 (0.160)	0.777 (0.000)	8.036 (0.466)	0.708
RFC_B	0.732 (0.000)	0.200 (0.000)	15.427 (0.000)	0.608
RFC_C	0.792 (0.000)	0.148 (0.000)	6.039 (0.000)	0.709

Coefficients are standardized, numbers in brackets are the p-values.

Summary

- **Main Results of the Research**

  - As "Object-Oriented ness" $\downarrow$,
    correlation of CK Metrics Suite $\uparrow$
  - As "Object-Oriented ness" $\downarrow$,
    RFC = NLM + NRM $\Rightarrow$ RFC = WMC + CBO

- **Benefit to NASA**

  - Adhering to good object-oriented principles produces higher quality software
  - The simple equation RFC = WMC + CBO evaluates object-oriented constructs